ABSTRACT

Heuristic search is a fundamental problem-solving method in artificial intelligence. The main limitation of search is its computational complexity which can be overcome by parallel implementation of the algorithms. Distributed tree search and Parallel window search are two of the approaches to parallelizing search algorithms.

We are developing an algorithm called HyPS (Hybrid Parallel Search) which is a combination of distributed tree search (RKR87) and parallel window search (PK91). In the HyPS, the set of processors is divided into clusters. Each cluster searches the same space, but uses a unique cost threshold. Within each cluster, processors are given unique portions of the search space to expand. Each cluster, adopts the rule of parallel windows, but distributed tree search is performed within a cluster. First, the initial state is expanded generating distinct subtrees. The number of distinct subtrees generated is equal to the number of processors in a cluster. Each processor in a cluster receives a subtree on which IDA* search is performed. The first cluster is given the heuristic estimate to reach the goal. All remaining clusters are given incrementally larger thresholds. The algorithm finds a first solution or an optimal solution. Many of the processors will be idle when searching for an optimal solution. Load balancing within a cluster will overcome the idling of processors within a cluster. HyPS is implemented on Connection Machine 5. The parallel algorithm is tested for different cluster sizes to find the first solution and the optimal solution. We are interested in finding which cluster size yields the best solution.

The domain of testing is the Fifteen Puzzle problem and the robot path planning problem. The average speedup and efficiency for the Fifteen Puzzle Problem instances are shown in the first table. Also the average speedup and efficiency for the Robot Planning Problem instances are shown in the second table. The initial testing shows marked improvement over serial and parallel algorithms.

References
